Pengamanan Sistem *Login* Aplikasi Menggunakan Protokol ID *Based* Diffie-Hellman *Key Agreement*

Aprita Danang Permana, S.ST
Jl. Harsono RM No. 70, Ragunan, Pasar Minggu, Jakarta Selatan 12550
aprita.danang@lemsaneg.go.id

Abstrak—Login **merupakan tahapan awal untuk kita dapat menjalankan sebuah aplikasi baik *email*, *chat*, ataupun aplikasi catatan pribadi seseorang. Dengan adanya *login*, dimungkinkan bahwa orang lain tidak dapat melakukan akses terhadap aplikasi tersebut. Biasanya sistem *login* hanya terdiri *username* dan *password*, namun jika keduanya dapat diketahui oleh orang lain, maka aplikasi menjadi tidak aman, bahkan orang lain tersebut dapat sesuka hati mengubah, menyimpan data-data yang ada pada aplikasi. Beberapa metode pengamanannya banyak ditawarkan oleh *administrator server*, salah satu yang paling sederhana yaitu menggunakan Hash Function (SHA-1, MD5). Namun hal tersebut masih belum cukup aman untuk sistem penyimpanan akun *login*. Dengan hal tersebut, pada makalah ini menawarkan solusi pengamanan sistem *login* dengan menggunakan protokol ID *based* Diffie Hellman *Key Agreement*. Protokol tersebut melakukan komputasi berdasarkan pada perkalian grup bilangan bulat yang memanfaatkan identitas sebagai *input* pembangkit kuncinya. Dari hasil implementasi yang telah dilakukan dapat meningkatkan keamanan sistem *login* tersebut.**

Kata kunci—Login, Diffie-Hellman, Secure Login, Key Agreement Protocol.

I. PENDAHULUAN

Saat sekarang ini terdapat berbagai jenis aplikasi yang dikembangkan. Baik ditujukan untuk sosial media, maupun aplikasi yang digunakan untuk penelitian. Sebagai contoh aplikasi Facebook yang telah banyak digemari oleh banyak orang, lalu dikembangkan hingga dapat digunakan melalui perangkat *mobile*. Setiap aplikasi yang dikembangkan terdapat salah satu bagian sederhana namun dapat membuat data-data di dalam aplikasi menjadi rusak bahkan hilang. Bagian sederhana tersebut yaitu sistem *login* aplikasi. Dengan adanya sistem *login* aplikasi, pengguna dapat mengamankan data-data yang ada di dalam aplikasi menggunakan akun yang sesuai dengan identitas pengguna tersebut.

Dengan kondisi seperti hal tersebut, sistem *login* menjadi bagian penting untuk dilakukan pengamanan sistemnya. Baik secara penyimpanan maupun penggunaanya. Hal penyimpanan sistem *login* menjadi tanggung jawab administrator sistem, sedangkan penggunaannya menjadi tanggung jawab pengguna itu sendiri. Dalam hal penyimpanan, akun *login* harus dilakukan sistem pengamanan yang tinggi, karena jika terjadi kebocoran, maka kerugian akan berdampak pada pemilik aplikasi dan juga pemilik akun. Pemilik aplikasi dapat dituntut karena kelalaian dari kebocoran akun pengguna tersebut.

Oleh karena itu, diperlukan sebuah sistem pengamanan *database* akun *login* pengguna agar dapat menjaga integritas data tersebut. Dalam makalah ini dibahas mengenai pemanfaatan protokol ID *Based* Diffie-Hellman *Key Agreement* (DHKA) sebagai pembangkit nilai rahasia dengan memanfaatkan identitas dari pengguna.

II. PENDAHULUAN

A. Protokol *Key Agreement*

Suatu protokol adalah serangkaian langkah yang melibatkan dua pihak atau lebih dan dirancang untuk menyelesaikan suatu tugas. Suatu protokol memiliki beberapa karakteristik antara lain yaitu :

a. setiap pihak yang terlibat dalam protokol harus mengetahui terlebih dahulu mengenai protokol dan seluruh langkah yang akan dilaksanakan;
b. setiap pihak yang terlibat dalam protokol harus menyetujui untuk mengikutinya;
c. protokol tidak boleh menimbulkan kerancuan;
d. protokol harus lengkap (langkah-langkah harus lengkap dari awal hingga akhir, tidak ada yang terlewat).

Protokol kriptografi adalah suatu protokol yang menggunakan kriptografi. Protokol ini melibatkan sejumlah algoritma kriptografi, namun secara umum tujuan protokol lebih dari sekedar kerahasiaan. Penggunaan kriptografi dalam sebuah protokol terutama ditujukan untuk mencegah ataupun mendeteksi adanya *eavesdropping* dan *cheating* [1].

Salah satu jenis protokol kriptografi yaitu *key agreement protocol* atau protokol pertukaran kunci. Protokol *key agreement* merupakan salah satu protokol kriptografi yang menyediakan layanan kriptografi untuk menghasilkan kunci [2]. Kunci yang dihasilkan merupakan kunci yang disetujui bersama oleh pihak-pihak yang berkomunikasi. Kunci tersebut merupakan kunci rahasia yang akan digunakan sebagai kunci simetrik untuk komunikasi yang aman [1].

B. Skema ID *Based* DHKA

Diffie-Hellman *Key Agreement* (DHKA) merupakan salah satu protokol *key agreement* yang didasarkan pada perkalian grup bilangan bulat. DHKA dikenalkan oleh Whitfild Diffie

dan Martin Hellman pada tahun 1976. Gambaran skema protokol DHKA dapat dilihat pada Tabel I.

TABEL I. SKEMA PROTOKOL DHKA

Parameter publik : p, g	
Alice : - Memilih kunci privat a : {1, p-1} - Menghitung nilai publik Alice : $t_a = g^a \bmod p$ - Mengirimkan nilai t_a ke Bob	Bob : - Memilih kunci privat b : {1, p-1} - Menghitung nilai publik Bob : $t_b = g^b \bmod p$ - Mengirimkan nilai t_b ke Alice
Komputasi nilai *shared secret* : Z	
Alice : - Mempunyai kunci publik Bob : t_b - Menghitung nilai $Z = t_b^a \bmod p$	Alice : - Mempunyai kunci publik Alice : t_a - Menghitung nilai $Z = t_a^b \bmod p$
$Z = t_b^a \bmod p$ $Z = (g^b \bmod p)^a \bmod p$ $Z = (g^b)^a \bmod p$ $Z = g^{ba} \bmod p$ $Z = g^{ab} \bmod p$ $Z = (g^a \bmod p)^b \bmod p$ $Z = t_a^b \bmod p$	

Skema ID Based DHKA merupakan pengembangan protokol DHKA yang memanfaatkan identitas publik pengguna dalam komputasi nilai rahasianya. Skema ini terdiri dari 3 (tiga) langkah, yaitu tahap *setup*, *compute key*, dan *secret value*. Sebelum melakukan ketiga tahapan tersebut, pengguna diharuskan melakukan registrasi. Tahap *setup* terdiri dari tahapan pemilihan parameter publik oleh sistem, lalu pembangkitan kunci privat dari identitas tersebut. Tahap *compute key* dilakukan komputasi kunci publik identitas tersebut. Dari identitas privat dan publik tersebut dihitung pada tahap *secret value*.

Misalkan salah seorang pengguna (nama: Alice) mendaftarkan identitasnya untuk mendapatkan akun sebuah aplikasi:

TABEL II. SKEMA PROTOKOL

Alice memasukkan data registrasi : - Nama lengkap - Alamat - Alamat *email* - *Username* - *Password*
Setup
- Sistem memilih nilai publik p dan g - Hitung nilai privat (S_{ID})
Q_{id1} = Hash(*username*); Q_{id2} = Hash(Hash(*username*));
Compute Key
- Hitung nilai publik $S_{ID1} = g\ Q_{id2} \bmod p$ $S_{ID2} = g\ Q_{id1} \bmod p$
Secret Value
$Z = S_{ID2}\ Q_{id1} \bmod p$

III. IMPLEMENTASI

A. Simulasi Protokol DHKA

Pada sub bahasan ini, akan dijelaskan mengenai simulasi protokol Diffie-Hellman menggunakan bahasa pemrograman Java [3].

a. Pembangkitan Nilai Publik p dan g

Pembangkitan nilai publik p dan g menggunakan tipe data BigInteger dengan fungsi probablePrime.

```
r = new Random();
BigInteger p = BigInteger.probablePrime(bitLength, r);
BigInteger g = BigInteger.probablePrime(bitLength, r);
```

dengan "bitLength" merupakan panjang bit, "r" merupakan bilangan random.

b. Pembangkitan Kunci Privat Alice dan Bob (Misal Merupakan User yang Hendak Berkomunikasi)

Pembangkitan kunci privat masing-masing pengguna menggunakan identitas sebagai masukkan pembangkit kunci. Pembangkitan dilakukan dengan konversi identitas yang berupa String diubah menjadi tipe data BigInteger yang memanfaatkan fungsi hash SHA1[5].

```
// Pembangkitan kunci privat pengirim
String idPengirim = GenerateSHA1.SHA1(identitas1.getText().toString());
BigInteger SID1 = hexToBigInt(idPengirim, m);  // secret alice

// Pembangkitan kunci privat penerima
String idPenerima = GenerateSHA1.SHA1(identitas2.getText().toString());
BigInteger SID2 = hexToBigInt(idPenerima, m);  // secret bob
```

c. Pembangkitan *Shared Secret*

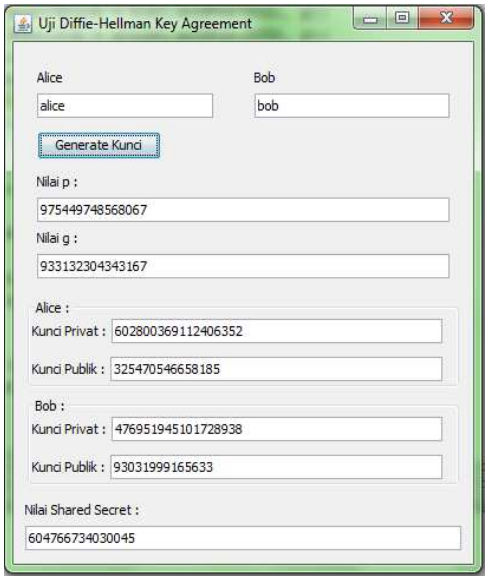
Sebelum dilakukan komputasi *shared secret*, dilakukan komputasi kunci publik masing-masing pengguna :

```
// Pembangkitan key agreement
BigInteger alice = g.modPow(SID1, p);
BigInteger bob   = g.modPow(SID2, p);
```

Lalu dilanjutkan dengan komputasi *shared secret*.

```
// Kunci
BigInteger kunciAlice = bob.modPow(SID1, p);
BigInteger kunciBob = alice.modPow(SID2, p);
```

Simulasi protokol Diffie-Hellman *Key Agreement* dapat dilihat pada Gambar 1.



Gambar 1. Simulasi Protokol

Pada Gambar 1, dijelaskan bahwa *user* pertama dan kedua menggunakan identitas nama sebagai pembangkit kunci privat. Setelah dibangkitkan kunci privat dan kunci publik, dihasilkan nilai *shared secret* yang sama diantara keduanya.

B. Implementasi Sistem

Implementasi sistem pengamanan sistem *login* menggunakan bahasa pemrograman Java. Penyimpanan akun dalam database menggunakan MySql.

Implementasi sistem ini mempunyai 2 (dua) fitur utama yaitu *Login form* dan *Registrasi form*. Pada fitur *Login form*, pengguna diharuskan memasukkan *username* dan *password* yang telah diregistrasikan sebelumnya. Jika belum mempunyai akun, pengguna tersebut diharuskan melakukan registrasi pada fitur *Registrasi form*. Pada *Registrasi form*, pengguna memasukkan data-data pribadi seperti nama lengkap, alamat tempat tinggal, alamat *email*, *username*, dan *password*.

Gambar 2. *Login Form*

Gambar 3. *Registrasi Form*

Setelah pengguna telah memasukkan identitasnya, maka sistem melakukan tahapan *setup* yaitu menghitung nilai privat dari pengguna tersebut. Lalu dilanjutkan dengan tahapan *compute key*. Kedua tahapan tersebut menggunakan *input* berupa *username*, nama, dan *password*. Hal ini karena berkaitan langsung dengan identitas pengguna yang nantinya akan dijadikan otentikasi terhadap pengguna tersebut jika terjadi gangguan.

Pada *database* penyimpanan akun, terdapat 5 (lima) kolom yang harus diisi. Implementasi *database* menggunakan MySQL[4]. Dengan 2 (dua) diantaranya merupakan hasil komputasi protokol ID *Based* DHKA.

nama	alamat	email	username	password
s Aprita	Jakarta	12@gmail.com	19750914702405361675126048245O	126f5ba376a73ddee15c5e86714d9fec705b1f64
s Danang	Banyumas	12@gmail.com	18742524179273089752097942364O	5ed03fbd3d566ef8f3acc4be95d5884f51abae9f

Gambar 4. *Database* Penyimpanan Akun

```
String userDHKA = GeneratePotokol.Diffie_Hellman(username, namaHash);
String pwdDHKA = GeneratePotokol.Diffie_Hellman(pwd1, nama);

String sql = "insert into tb_securelogin(nama,alamat,email,username,password) values"
      + "('" + nama + "', '" + alamat + "', '" + email + "', '" + userDHKA + "', SHA1('" + pwdDHKA + "'))";
```

IV. PENGUJIAN DAN ANALISIS

A. Pengujian Fungsional

Pada pengujian fungsional, dilakukan pengujian pada fitur utama sistem *login* ini.

a. Fitur Login

Pada fitur ini dilakukan proses pemeriksaan koneksi antara aplikasi dengan *database server* (Mysql). Jika *database server* dalam keadaan *offline* maka akan muncul notifikasi bahwa koneksi gagal.

Gambar 5. Pemeriksaan *Database Server*

Setelah proses pemeriksaan *database server*, dilanjutkan dengan proses pemeriksaan akun yang dimasukkan oleh pengguna dengan akun yang tersimpan pada *database server*. Jika benar, maka akan masuk ke dalam main menu aplikasi, jika salah maka akan muncul notifikasi kesalahan.

Gambar 6. Notifikasi Kesalahan Akun

b. Fitur Registrasi

Pada fitur registrasi, pengguna diharuskan mengisi data diri yang dijadikan identitas dan akun untuk melakukan *login* aplikasi.

Gambar 7. Form Registrasi Pengguna

Pengujian dilakukan dengan mencoba mengosongkan salah satu field pada form tersebut, maka akan muncul notifikasi kesalahan.

Gambar 8. Notifikasi Kesalahan

Dari pengujian fungsional yang telah dilakukan, dapat disimpulkan pada Tabel III.

TABEL II. HASIL PENGUJIAN FUNGSIONAL

No.	Fitur	Pengujian	Hasil
1.	Login	Akun benar	Masuk main menu aplikasi
		Akun salah	Muncul notifikasi kesalahan
2.	Registrasi	*Field* kosong	Registrasi berhasil
		Field terisi semua	Muncul notifikasi kesalahan
		Field password < 8 karakter	Muncul notifikasi panjang password > 8 karakter
		Field password ≠ konfirmasi *password*	Muncul notifikasi Password dan Konfirmasi password tidak sama

B. Pengujian Kecepatan

Pada pengujian kecepatan, dilakukan pengujian terhadap komputasi protokol Diffie-Hellman pada saat melakukan komputasi kunci. Spesifikasi perangkat yang digunakan untuk pengujian sebagai berikut :

a. Sistem operasi : Windows 7
b. Intel(R) Core(TM) i7 Q720 @ 1.60GHz 1.60 GHz

Pengujian dilakukan dengan batas panjang bit untuk nilai publik p dan g yaitu dengan panjang 10 bit, 50 bit, 100 bit, 150 bit, 500 bit, dan 1000 bit. Hasil pengujiannya dapat dilihat pada Tabel IV.

TABEL IV. HASIL PENGUJIAN KECEPATAN KOMPUTASI

No.	Panjang bit	Percobaan	Waktu (ms)
1.	50 bit	1	21
		2	22
		3	41
		4	25
		5	18
		Rata-rata	25,4
2.	75 bit	1	40
		2	35
		3	30
		4	58
		5	51
		Rata-rata	42,8
3.	100 bit	1	87
		2	44
		3	41
		4	51
		5	86
		Rata-rata	61,8

4.	150 bit	1	133
		2	171
		3	202
		4	177
		5	100
		Rata-rata	156,6
5.	500 bit	1	319
		2	910
		3	2314
		4	1931
		5	917
		Rata-rata	1278,2
6.	1000 bit	1	8241
		2	19656
		3	9559
		4	9641
		5	10384
		Rata-rata	11496,2

C. Analisis

Berdasarkan pengujian yang telah dilakukan, fitur pada sistem *login* dapat berfungsi sesuai dengan fungsinya yaitu fungsi *Login* dan fungsi Registrasi. Implementasi protokol diffie-hellman membutuhkan waktu komputasi rata-rata 11.496,2 detik dengan pembangkitan panjang karakter 1000 bit. Semakin panjang karakter yang dibangkitkan, maka proses komputasi akan semakin meningkat.

Berdasarkan hasil implementasi, terbukti bahwa implementasi protokol diffie-hellman dapat mengamankan identitas (*username* dan *password*) yang ada pada *database* penyimpanan identitas *user*. *Username* dan *password* pengguna tidak dapat dibaca dengan metode pembacaan biasa. Dengan hal ini, pihak yang tidak berwenang tidak mendapatkan identitas dari pengguna, untuk dapat dilakukan perubahan ataupun perusakan pada aplikasi dari pengguna.

nama	alamat	email	username	password
s Aprita	Jakarta	12@gmail.com	19750914702405361675126048245O	126f5ba376a73ddee15c5e86714d9fec705b1f64
s Danang	Banyumas	12@gmail.com	18742524179273089752097942364O	5ed03fbd3d566ef8f3acc4be95d5884f51abae9f

V. SIMPULAN

Dari penelitian yang telah dilakukan dapat disimpulkan sebagai berikut :

1. Protokol ID Based Diffie-Hellman *Key Agreement* dapat diterapkan sebagai pembangkit nilai rahasia untuk mengamankan sistem *login* pada akun pengguna.
2. Pembangkitan nilai rahasia memanfaatkan identitas publik pengguna yaitu nama, *username*, dan *password*. Ketiganya dilakukan komputasi untuk menghasilkan nilai rahasia sebagai penyimpanan akun *login* aplikasi pengguna.

Dari pengujian fungsional yang dilakukan, fitur pada sistem *login* yang dibuat telah dapat berjalan sesuai dengan tujuannya. Sedangkan dari segi kecepatan komputasi, proses ID *Based* Diffie-Hellman *Key Agreement* membutuhkan 11.4 detik untuk penggunaan panjang kunci 1000bit.

DAFTAR PUSTAKA

[1] Menezes, Alfred J.. Paul C. Van Oorschot. Scott A. Vanstone. 1997. Handbook of Applied Cryptography. CRC press LLC. Boca Raton.

[2] Boyd, Colin, Anish Mathuria. 2003. Protocols for Authentication and Key Establishment. Springer.

[3] Sharon Zakhour et al. The Java Tutorial Fourth Edition. http://docs.oracle.com/javase/tutorial/

[4] Setiyadi, Didik. 2010. Modul Praktikum Perancangan Basis Data SQL Server, STMIK Eresha.

[5] Cay Horstmann. 2010. Java Concepts 6th Edition. John Wiley & Sons.